arXiv:2002.05505v5 [cs.LG] 14 Aug 2020

Assessment Modeling: Fundamental Pre-training Tasks for Interactive Educational Systems

Youngduck Choi[1,2], Youngnam Lee[1], Junghyun Cho[1], Jineon Baek[1,3], Dongmin Shin[1], Seewoo Lee[1,4] Jonghun Shin[1], Chan Bae[1,4], Byungsoo Kim[1], Jaewe Heo[1]
[1]Riiid! AI Research, [2]Yale University, [3]University of Michigan, [4]UC Berkeley
{youngduck.choi, yn.lee, jh.cho, jineon.baek, dm.shin
seewoo.lee, jonghun.shin, chan.bae, byungsoo.kim, jwheo}@riiid.co

ABSTRACT

Interactive Educational Systems (IESs) have developed rapidly in recent years to address the issue of quality and affordability in education. Like many other domains in Artificial Intelligence (AI), there are specific tasks in the field of AI in Education (AIEd) for which labels are scarce and expensive. For instance, information about exam scores and grades are essential to understanding a student's educational progress and is a key factor affecting social outcomes. However, unlike interactive features automatically collected by IESs, obtaining the labels is costly as they are often generated outside the IES. Other examples of scarce labels include data on course dropout and review correctness. While this data is automatically recorded by IESs, they tend to be few in number as the events occur sporadically in practice. A common way of circumventing the label-scarce problems is via the pre-train/fine-tune method, where a model is trained in a relevant auxiliary task with a large amount of data before the main task. Accordingly, existing works pre-train a model to learn representations of the contents of learning items (e.g. exercises). However, such methods fail to utilize the full range student interaction data available and do not model student learning behavior.

To this end, we propose Assessment Modeling, a class of fundamental pre-training tasks for general IESs. An assessment is a feature of student-system interactions which can serve as a pedagogical evaluation. Examples include the correctness of a student's answer and the time taken for the student to answer. Assessment Modeling is the prediction of assessments conditioned on the surrounding context of interactions. Although it is natural to pre-train on interactive features available in large amounts, limiting the prediction targets to assessments focuses the tasks' relevance to the label-scarce educational problems and reduces less-relevant noise.

To the best of our knowledge, this is the first work investigating appropriate pre-training methods for predicting educational features from student-system interactions. Training scheme in Assessment Modeling poses various challenges. For example, one should consider which assessments to use in pre-training for a specific label-scarce educational problem. Also, an asymmetry among a set of available features, assessments being masked, and assessments being predicted in each time step is a nontrivial issue. While the effectiveness of different combinations of assessments is open for exploration, we suggest Assessment Modeling as a first-order guiding principle for selecting proper pre-training tasks for label-scarce educational problems.

Keywords

Artificial Intelligence in Education, Label-scarcity, Pre-training, Assessment Modeling

1. INTRODUCTION

An Interactive Educational System (IES) interacts with students to assess them and to design individualized optimal learning paths. IESs automatically collect observations of student behaviors at scale, and can thus power data-driven approaches for many Artificial Intelligence in Education (AIEd) tasks. This has the potential to greatly improve the quality of IESs. However, there are important tasks where a lack of data prevents relevant models from attaining their full potential. A possible case is when the task depends on labels external to IESs. For example, the labels for a grade prediction model in an in-class IES may be generated at the end of class and outside the IES. In some cases, the data points are generated and collected by IESs but simply occur sporadically in practice. For instance, data on students reviewing previously solved exercises may be scarce compared to data on students solving unseen exercises since students tend to invest more time in solving new exercises than in reviewing solved ones.

To circumvent the lack of data, transfer learning has been explored in the AIEd literature [15, 9]. However, previously explored methods can work only when the available data and the task to complete have the same form. To bypass this restriction, we propose a methodology in the pre-train/fine-tune paradigm. In this paradigm, a model is first pre-trained in an unsupervised auxiliary task for which data is abundant. Then, the model is slightly modified to match the main task and trained (fine-tuned) with possibly scarce data. This approach has seen success in other subfields of AI including Natural Language Processing (NLP), Computer Vision, and

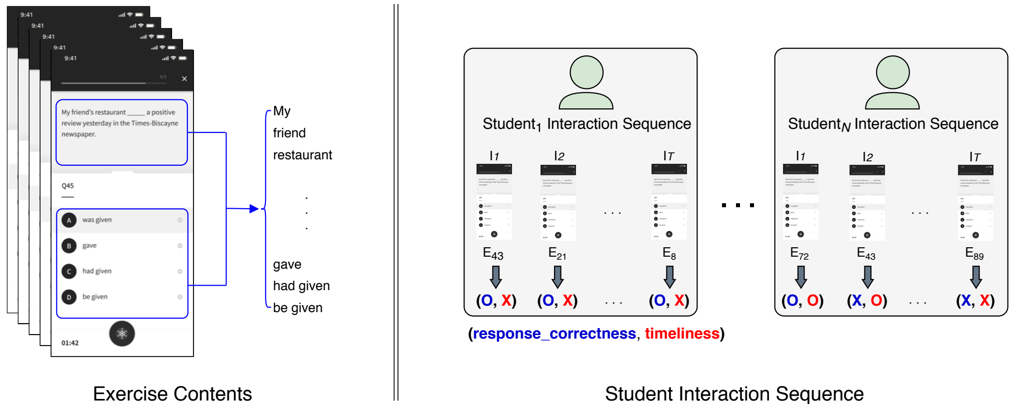

Figure 1: Comparison between content-based and interaction-based approaches. Content-based approaches learn representations of contents in learning items (e.g. exercises). On the other hand, interaction-based approaches model student learning behaviors in interactive educational systems.

Motion Planning [7, 27, 26]. Following this line of inquiry, content-based pre-training methods have been studied by [13, 28, 30]. However, student interactions are not considered and the work was limited to capturing the content of learning materials. Accordingly, they do not make use of the information carried by the learning behavior of students using IESs.

In this paper, we propose Assessment Modeling, a class of fundamental pre-training tasks for general IESs. Here, an assessment is any feature of student-system interactions which can act as a criterion for pedagogical evaluation. Examples of assessments include the time a student spends on each exercise and the correctness of a student response to a given exercise. While there is a wide range of interactive features available, we narrow down the prediction targets to assessments to focus on the information most relevant to label-scarce educational problems. Also, most data on assessments is available in large amounts as they are automatically collected by IESs. Inspired by the recent success of bidirectional representations in NLP domain [8], we develop an assessment model using a deep bidirectional Transformer encoder. In the pre-training phase, we randomly select a portion of entries in a sequence of interactions and mask the corresponding assessments. Then, we train a deep bidirectional Transformer encoder-based assessment model to predict the masked assessments conditioned on the surrounding interactions. After pre-training, we replace the last layer of the model with a layer corresponding to each downstream task, and all parameters of the model are then fine-tuned to the downstream tasks.

We empirically evaluate the use of Assessment Modeling as pre-training tasks. Our experiments are on *EdNet* [4], a large-scale dataset collected by an active mobile education application, *Santa*, which has 1M users as well as 72M response data points gathered since 2016. The results show that Assessment Modeling provides a substantial performance improvement in downstream AIEd tasks. In particular, we obtain an improvement of 14.16 mean absolute error and 0.014 accuracy from the previous state-of-the-art model for exam score and review correctness prediction respectively.

In summary, our contributions are as follows:

- We propose Assessment Modeling, a class of fundamental pre-training tasks for general IESs.
- We give formal definitions of Knowledge Tracing and Assessment Modeling in a form that is quantifiable and objective. We also provide examples in the context of a particular IES design.
- Inspired by the recent success of bidirectional representation in NLP domain [8], we propose an assessment model using a deep bidirectional Transformer encoder.
- We report empirical results showing that using Assessment Modeling as pre-training tasks achieves an improvement of 14.16 mean absolute error and 0.014 accuracy compared to the previous state-of-the-art model for exam score prediction and review correctness prediction respectively.

2. RELATED WORKS

2.1 Artificial Intelligence in Education

AIEd supports education through different AI technologies, including machine learning and deep learning, to "promote the development of adaptive learning environments and other AIEd tools that are flexible, inclusive, personalised, engaging, and effective" [20]. There is now a large body of work on the development of AI models for AIEd tasks, including knowledge tracing [5, 25], question analysis [30, 18], student score/grade prediction [16, 22, 1, 24], educational content recommendation [19] and many more. These models allow an IES to keep track of the evolving state of each student through constant feedback. This knowledge can be used to provide a tailored learning experience suited to individual students.

2.2 Pre-training Methods in Education

Pre-training is the act of training a model to perform an unsupervised auxiliary task before using the trained model to perform the supervised main task [10]. Pre-training has been shown to enhance the performance of models relative to existing models in various fields including NLP [7], Computer

Vision [27], Speech Recognition [26], and Medical Science [3]. Pre-training techniques have been also applied to educational tasks with substantial performance improvements. For example, [15] predicts whether a student will graduate or not based on studentsâĂŹ general academic information such as SAT/ACT scores or courses taken during college. They predict the graduation of 465 engineering students by first pre-training on the data of 6834 students in other departments using the TrAdaBoost algorithm [6]. [9] suggests two transfer learning methods, Passive-AE transfer and Active-AE transfer, to predict student dropout in Massive Open Online Courses (MOOCs). Their experimental results show that both methods improve the prediction accuracy, with Passive-AE transfer more effective for transfer learning across the same subject and Active-AE transfer more effective for transfer learning across different subjects.

Most of the pre-training methods used in interactive educational system are NLP tasks with training data produced from learning materials. For example, the short answer grading model suggested in [28] uses a pre-trained BERT model to ameliorate the limited amount of student-answer pair data. They took a pre-trained, uncased BERT-base model and fine-tuned it on the ScientsBank dataset and two psychology domain datasets. The resulting model outperformed existing grading models.

Test-aware Attention-based Convolutional Neural Network (TACNN) [13] is a model that utilizes the semantic representations of text materials (document, question and options) to predict exam question difficulty (i.e. the percentage of examinees with wrong answer for a particular question). TACNN uses pre-trained word2vec embeddings [21] to represent word tokens. By applying convolutional neural networks to the sequence of text tokens and an attention mechanism to the series of sentences, the model quantifies the difficulty of the question.

QuesNet [30] is a question embedding model pre-trained with the context information of question data. Since existing pre-training methods in NLP are unsuited for heterogeneous data such as images and metadata in questions, the authors suggest the Holed Language Model (HLM), a pre-training task, in parallel to BERTâĂŹs masking language model. HLM differs from BERT's task, however, because it predicts each input based on the values of other inputs aggregated in the Bi-LSTM layer of QuesNet, while BERT masks existing sequences at random. Also, QuesNet introduces another task called Domain-Oriented Objective (DOO), which is the prediction of the correctness of the answer supplied with the question, to capture high-level logical information. QuesNet adds a loss for each of HLM and DOO to serve as its final training loss. Compared to other baseline models, QuesNet shows the best performance in three downstream tasks: knowledge mapping, difficulty estimation, and score prediction.

3. ASSESSMENT MODELING

3.1 Formal Definition of Assessment Modeling

Recall that Knowledge Tracing is the task of modeling a studentâĂŹs knowledge state based on the history of their learning activities. Although Knowledge Tracing is widely considered a fundamental task in AIEd and has been studied extensively, there is no precise definition in the literature. In this subsection, we first define Knowledge Tracing in a form that is quantifiable and objective for a particular IES design. Subsequently, we introduce a definition of Assessment Modeling that addresses the educational values of the label being predicted.

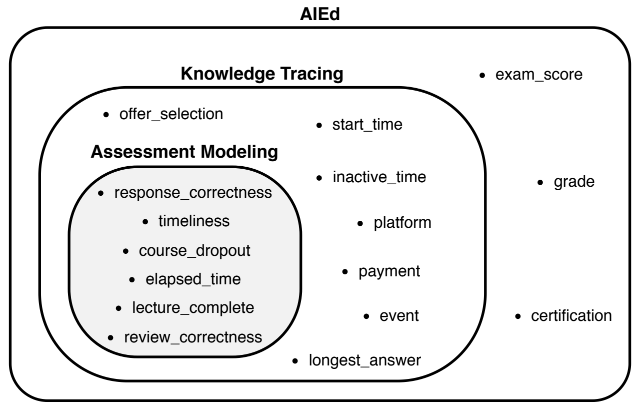

Figure 2: The features predicted in general AIEd tasks, Knowledge Tracing and Assessment Modeling. Assessment Modeling predicts the distribution of assessments, the subset of interactive features which can act as pedagogical evaluation criteria. Note that predicting an exam score (`exam_score`), a grade (`grade`) and whether a student will pass to get a certificate (`certification`) are tasks outside Knowledge Tracing.

A learning session in an IES consists of a series of interactions $[I_1, \ldots, I_T]$ between a student and the system, where each interaction $I_t = \{f_t^1, \ldots, f_t^n\}$ is represented as a set of features f_t^k automatically collected by the system. The features represent diverse aspects of learning activities provided by the system, including the exercises or lectures being used, and the corresponding student actions. Using the same notation, we define Knowledge Tracing and Assessment Modeling as follows:

Definition 1. *Knowledge Tracing* is the task of predicting a feature f_t^k of the student in the t'th interaction I_t given the sequence of interactions $[I_1, \ldots, I_T]$. That is, the prediction of

$$p(f_t^k | \{I_1, \ldots, I_{t-1}, I_t, I_{t+1}, \ldots, I_T\} \setminus \Gamma(f_t^k)) \qquad (1)$$

for some Γ, where $\Gamma(f)$ is the set of features that should be masked when the feature f is guessed. This is to mask input features not available at prediction time, so that the model does not 'cheat' while predicting f.

This definition is compatible with prior uses of the term in works on Knowledge Tracing models [25, 31, 14, 23]. Although a common set-up of Knowledge Tracing models is to predict a feature conditioned on only past interactions, we define Knowledge Tracing as a prediction task that can also be conditioned on future interactions to encompass the recent successes of bi-directional architectures in Knowledge Tracing [17].

Example 1 (**Knowledge Tracing**)**.** A typical instance of a Knowledge Tracing task might be response correctness prediction, where the interaction $I_i = \{e_i, r_i\}$ consists of an exercise e_i given to a student, and the correctness r_i of the student's corresponding response [25, 31, 14, 23, 17]. In this setup, only the response correctness r_T of the last interaction I_T is predicted and the features related to r_T are masked. Following our definition of Knowledge Tracing, the task can be extended further to predict diverse interactive features such as:

- `offer_selection`: Whether a student accepts studying the offered learning items.
- `start_time`: The time a student starts to solve an exercise.
- `inactive_time`: The duration for which a student is inactive in a learning session.
- `platform`: Whether a student responds to each exercise on a web browser or a mobile app.
- `payment`: Whether a student purchases paid services.
- `event` : Whether a student participates in application events.
- `longest_answer`: Whether a student selected the answer choice with the longest description.
- `response_correctness`: Whether a student responds correctly to a given exercise.
- `timeliness`: Whether a student responds to each exercise under the time limit recommended by domain experts.
- `course_dropout` : Whether a student drops out of the entire class.
- `elapsed_time`: The duration of time a student takes to solve a given exercise.
- `lecture_complete`: Whether a student completes studying a video lecture offered to them.
- `review_correctness` : Whether a student responds correctly to a previously solved exercise.

In the aforementioned example, features like `response_correctness` and `timeliness` directly evaluates the educational values of a student interaction, while it is somewhat debatable whether `platform` and `longest_answer` are also capable of addressing such qualities. Accordingly, we define *assessments* and *Assessment Modeling* as the following.

Definition 2. An *assessment* a_t^k of the t'th interaction I_t is a feature of I_t which can act as a criterion for pedagogical evaluation. The collection $A_t = \{a_t^1, \ldots, a_t^m\}$ of assessments is a subset of the available features $\{f_t^1, \ldots, f_t^n\}$ of I_t. *Assessment Modeling* is the prediction of assessment a_t^k for some k from the interactions $[I_1, \ldots, I_T]$. That is, the prediction of

$$p(a_t^k | \{I_1, \ldots, I_{t-1}, I_t, I_{t+1}, \ldots, I_T\} \setminus \Gamma(a_t^k)) \qquad (2)$$

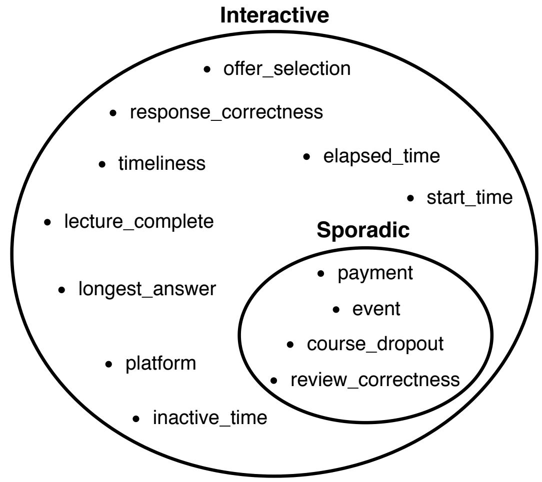

Figure 3: Not all interactive features are collected equally often. For example, `payment`, `event`, `course_dropout` and `review_correctness` are interactive features obtained more sporadically than other features. Among the sporadic interactive features, `course_dropout` and `review_correctness` are assessments.

Example 2 (**Assessments**)**.** Among the interactive features listed in Example 1, we consider `response_correctness`, `timeliness`, `course_dropout`, `elapsed_time`, `lecture_complete` and `review_correctness` to be assessments. For example, `response_correctness` is an assessment as whether a student responded to each exercise correctly provides strong evidence regarding the student's mastery of concepts required to solve the exercise. Also, `timeliness` also serves as an assessment since the amount of time it takes a student to respond to each exercise is expected to contain information about their proficiency in the skills and knowledge necessary to solve the exercise. Figure 2 depicts the relationship between assessments and general Knowledge Tracing features.

3.2 Assessment Modeling as Pre-training Tasks

In this subsection, we provide examples of important yet scarce educational features and argue that Assessment Modeling enables effective prediction of such features.

Example 3 (**Non-Interactive Educational Features**)**.** In many applications, an IES is often integrated as part of a larger learning process. Accordingly, the ultimate evaluation of the learning process is mostly done independently from the IES. For example, academic abilities of students are measured by course grades or standardized exams, and the ability to perform a complicated job or task is certified by professional certificates. Such labels are considered essential due to the pedagogical and social needs for consistent evaluations of student ability. However, obtaining these labels are often challenging due to their scarcity compared to that of features automatically collected from student-system interactions. We give the following examples (see Figure 2).

- `exam_score` : A student's score on a standardized exam.
- `grade` : A student's final grade in a course.

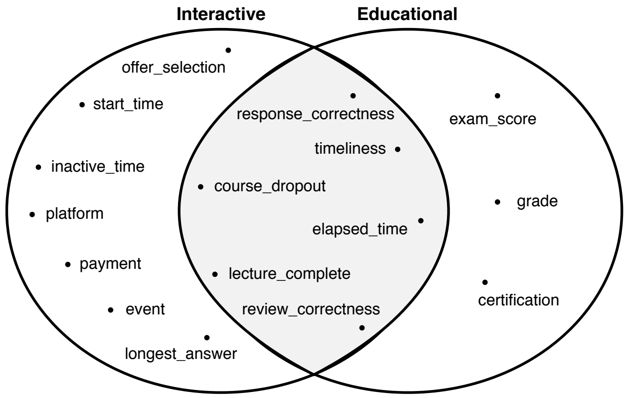

Figure 4: Assessment Modeling is an effective pre-training method for label-scarce educational problems. First, most assessments are available in large amounts as they are automatically collected from student-system interactions (Interactive). Also, assessments are selected to be relevant to educational progress, narrowing down the scope of prediction targets and reducing noise irrelevant to the problems (Educational).

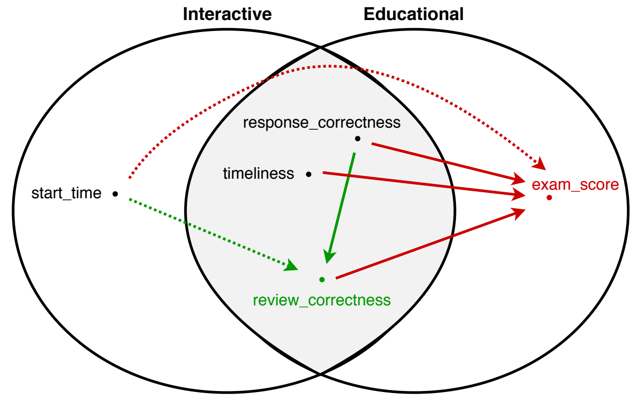

Figure 5: Possible pre-train/fine-tune scenarios. We may pre-train a model to predict `start_time`, `response_correctness`, `timeliness` and `review_correctness` and then train it to estimate `exam_score` (red). Likewise, a model pre-trained to predict `start_time` and `response_correctness` can be trained to predict `review_correctness` (green). However, pre-training to predict a non-educational interactive feature (non-assessment) like `start_time` is not effective in label-scarce educational problems (dotted line).

- `certification` : Professional certifications obtained by completion of educational programs or examinations.

Example 4 (Sporadic Assessments). All assessments are automatically collected by IESs, but some assessments are few in number as the corresponding events occur rarely in practice. For example, it is natural for students to invest more time in learning new concepts than reviewing previously studied materials. `course_dropout` and `review_correctness` are examples of *sporadic assessments* (Figure 3).

To overcome the aforementioned lack of labels, we consider the pre-train/fine-tune paradigm that leverages data available in large amounts to aid performance in tasks where labels are scarce. In this paradigm, a model is first trained in an auxiliary task relevant to the tasks of interest with label-scale data. Using the pre-trained parameters to initialize the model, the model is slightly modified to suit the task of interest, and then trained further (fine-tuned) on the main tasks. This approach has been successful in AI fields like NLP, computer vision and speech recognition [7, 27, 26]. Following this template, existing methods in AIEd pre-train on the contents of learning materials, but such methods do not capture student behavior and only utilize a small subset of features available from the data.

Instead, one may pre-train on different features automatically collected by IESs (see Figure 4). However, training on every available feature is computationally intractable and may introduce irrelevant noise. To this end, Assessment Modeling narrows down the prediction targets to assessments, the interactive features that also hold information on educational progress. Since multiple assessments are available, a wide variety of pre-train/fine-tune pairs can be explored for effective Assessment Modeling (see Figure 5). This raises the open-ended questions of *which* assessments to pre-train on for label-scarce educational problems and *how* to pre-train on multiple assessments.

Also, while the masking scheme for Assessment Modeling was inspired by masked language modeling [8], there is a key difference between the two approaches (Figure 6). In masked language modeling, the features available at a timestep are (the embeddings of) each word, the masked feature is the word at the timestep, and the target to predict is also the word at the timestep. That is, there is a symmetry in that the features that are available, the features being masked, and the features being predicted are all of the same nature. But that is not necessarily the case in Assessment Modeling. For example, suppose the features available at a given timestep are exercise id, category of the exercise, response correctness, and timeliness. A typical Assessment Modeling pre-training scheme may mask response correctness and timeliness, and predict just response correctness. This asymmetry raises the issue of precisely which features to mask and which features to predict, and the choices made will have to reflect the specific downstream task that Assessment Modeling is being used to prepare for. While we draw attention to this issue, it is outside the scope of this paper and we leave the details for future study.

In Section 4, we explore these issues for exam score (a non-interactive educational feature) and review correctness (a sporadic assessment) prediction. Experimental results support our claim modeling assessments are effective pre-training tasks for label-scarce educational problems.

3.3 Assessment Modeling with Deep Bidirectional Transformer Encoder

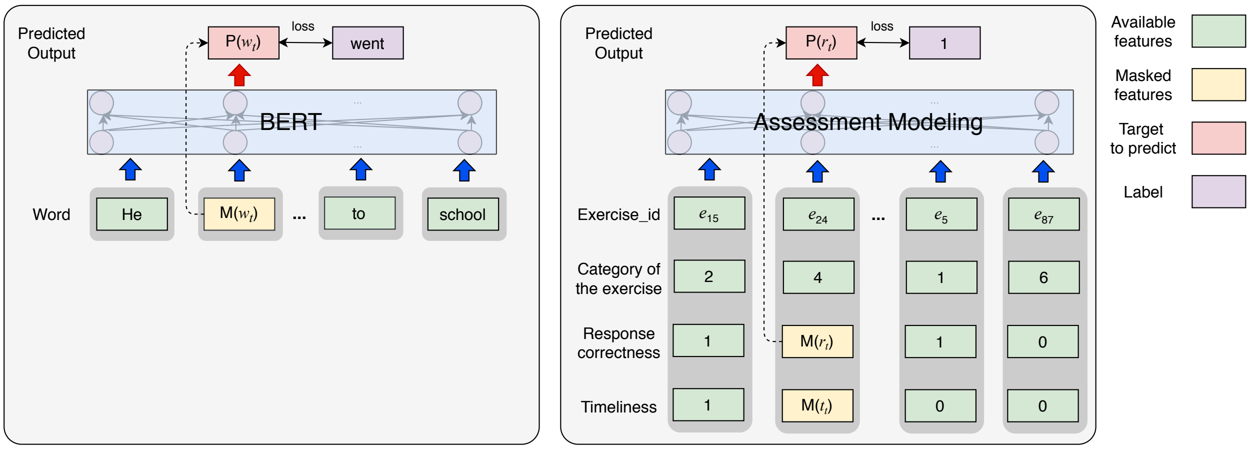

Figure 6: The issue of asymmetry arising in Assessment Modeling. In BERT's masked language modeling training scheme, features that are available, being masked and being predicted for each time step are all of the same nature. However, in Assessment Modeling, features being predicted can be a subset of features being masked, and features being masked can be a subset of features available in a specific time step.

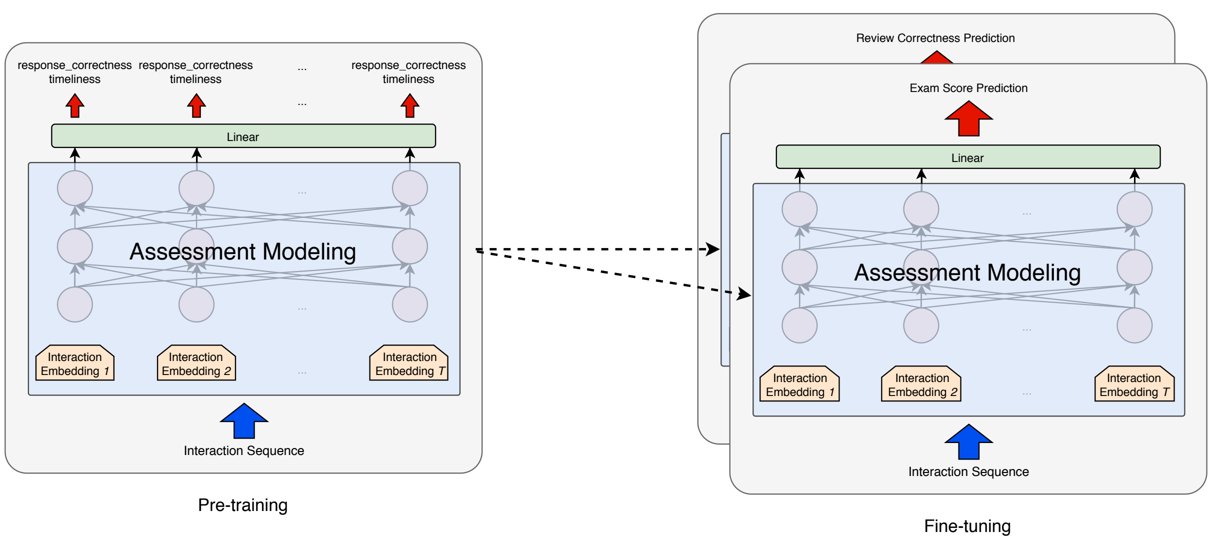

Figure 7: Proposed pre-train/fine-tune approach. In the pre-training phase, we train an assessment model to predict assessments conditioned on past and future interactions. After the pre-training phase, we fine-tune parameters in the model to predict labels in downstream tasks.

While there are several possible options for the architecture of the assessment model, we adopt the deep bidirectional Transformer encoder proposed in [8] for the following reasons. First, [23] showed that the self-attention mechanism in Transformer [29] is effective for Knowledge Tracing. The Transformer-based Knowledge Tracing model proposed in [23] achieved state-of-the-art performance on several datasets. Second, the deep bidirectional Transformer encoder model and pre-train/fine-tune method proposed in [8] achieved state-of-the-art results on several NLP tasks. While [8] conducted experimental studies in the NLP domain, the method is also applicable to other domains with slight modifications.

Figure 7 depicts our proposed pre-train/fine-tune approach. In the pre-training phase, we train a deep bidirectional Transformer encoder-based assessment model to predict assessments conditioned on past and future interactions. After the pre-training phase, we replace the last layer of the assessment model with a layer appropriate for each downstream task and fine-tune parameters in the whole model to predict labels in the downstream tasks. We provide detailed descriptions of our proposed assessment model in the following subsections.

3.3.1 Input Representation

The first layer in the assessment model maps each interaction to an embedding vector. First, we embed the following attributes:

- `exercise_id`: We assign a latent vector unique to each exercise.
- `exercise_category`: Each exercise has its own category tag that represents the type of the exercise. We assign a latent vector to each tag.
- `position`: The relative position t of the interaction I_t in the input sequence. We use the learned positional embedding from [11] instead of the sinusoidal positional encoding that is used in [29].

As shown in Example 1, an IES collects diverse interactive features that can potentially be used for Assessment Modeling. However, not only using all possible interactive features for Assessment Modeling is computationally intractable, there is no guarantee that the best results on downstream tasks will be achieved when all the features are used. For experimental studies, we narrow down the scope of interactive features to the ones available from an exercise-response pair, the simplest widely-considered interaction in Knowledge Tracing. In particular, we embed the following interactive features:

- `response_correctness`: The value is 1 if a student response is correct and 0 otherwise. We assign a latent vector corresponding to each possible value 0 and 1.
- `timeliness`: The value is 1 if a student responds within a specified time limit and 0 otherwise. We assign a latent vector corresponding to each possible value 0 and 1.

Let e_t be the sum of the embedding vectors of `exercise_id`, `exercise_category` and `position`. Likewise, let c_t and t_t be the embedding vectors of `response_correctness` and `timeliness` respectively. Then, the representation of interaction I_t is $e_t + c_t + t_t$.

3.3.2 Masking

Inspired by the masked language model proposed in [8], we use the following method to mask the assessments in a student interaction sequence. First, we mask a fraction C of interactions chosen uniformly at random. If the t-th interaction is chosen, we replace the corresponding input embedding with (1) $e_t + \texttt{mask}$ for a fraction M of the time and (2) $e_t + c_{\text{rand}} + t_{\text{rand}}$ for a fraction $(1 - M)$ of the time. Here `mask` is a learned vector that represents masking, and c_{rand} and t_{rand} are embedding vectors for assessments chosen uniformly at random from the sequence. We determine C and M through ablation studies in Section 4.

3.3.3 Model Architecture

After the interactions are embedded and masked accordingly, they enter a series of Transformer encoder blocks, each consisting of a multi-head self-attention layer followed by position-wise feed-forward networks. Every layer has input dimension d_{model}. The first encoder block takes the sequence of interactions $I_1, \ldots, I_T$ embedded in latent space and returns a series of vectors of the same length and dimension. For all $2 \leq i \leq N$, the i'th block takes the output of the $i-1$'th block as input and returns the series of vectors accordingly. We describe the architecture of each block as the following.

The multi-head self-attention layer takes a series of vectors, $X_1, \ldots, X_T$. Each vector is projected to latent space by projection matrices $W^Q, W^K \in \mathbb{R}^{d_{\text{model}} \times d_K}$ and $W^V \in \mathbb{R}^{d_{\text{model}} \times d_V}$:

$$\begin{aligned} Q &= [q_1, \ldots, q_T]^T = XW^Q \\ K &= [k_1, \ldots, k_T]^T = XW^K \\ V &= [v_1, \ldots, v_T]^T = XW^V \end{aligned} \tag{3}$$

Here $X = [X_1, \ldots, X_T]^T$ and each q_i, k_i and v_i are the query, key and value of X_i respectively. The output of the self-attention is then obtained as a weighted sum of values with coefficients determined by the dot products between queries and keys:

$$\text{Attention}(X) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V \tag{4}$$

Models with self-attention layers often use multiple heads to jointly attend information from different representative subspaces. Following this, we apply attention h times to the same query-key-value entries with different projection matrices for output:

$$\text{Multihead}(X) = \text{concat}(\text{head}_1, \ldots, \text{head}_T)W^O \tag{5}$$

Here, each head_i is equal to the output of self-attention in Equation 4 with corresponding projection matrices W_i^Q, W_i^K and W_i^V in Equation 3. We use the linear map W^O to aggregate each attention result.

After we compute the resulting value in Equation 5, we apply point-wise feed-forward networks to add non-linearity to the model. Also, we apply the skip connection [12] and layer normalization [2] to the output of the feed-forward networks.

Assume that the last encoder block returns the sequence $H = [H_1, \ldots, H_T]^T$ of vectors. For pre-training, the predictions to i'th assessments are made by applying a linear layer with the softmax activation function to H_i. The final output is the estimated probability distribution of four possible combinations of assessments A_t: (`response_correctness`$_t$, `timeliness`$_t$) $= (0,0)$, $(0,1)$, $(1,0)$ or $(1,1)$. The overall loss is defined to be

$$\mathcal{L} = \sum_{t=1}^{T} m_t \mathcal{L}_t$$

where $\mathcal{L}_t$ is the cross-entropy between the estimated distribution of A_t and the actual one-hot distribution of A_t. The value m_t is a flag that represents whether the t-th exercise is masked ($m_t = 1$) or not ($m_t = 0$). In other words, the total loss is the sum of cross-entropy losses for masked exercises.

The input embedding layer and encoder blocks are shared over pre-training and fine-tuning. For fine-tuning, we replace the linear layers applied to each H_i in pre-training with a single linear layer that combines all the entries of H to fit the output to downstream tasks.

4. EXPERIMENTS

4.1 Label-Scarce Educational Problems

We apply Assessment Modeling to exam score (a non-interactive educational feature) and review correctness (a sporadic assessment) prediction.

4.1.1 Exam Score Prediction

Exam Score prediction (ES) is the estimation of a student's scores in standardized exams, such as the TOEIC and the SAT, based on the student's interaction history with an educational system. ES is one of the most important tasks of AIEd, as standardized assessment is crucial for both the students and the educational system. Because a substantial amount of human effort is required to develop or take the tests, the number of data points available for exam score is considerably fewer than that of student interactions automatically collected by educational systems. By developing a reliable ES model, a student's universally accepted score can be estimated by an interactive educational system with considerably less effort. ES differs from student response prediction (e.g. the prediction of assessment `response_correctness`$_t$) because standardized tests are taken in a controlled environment with specific methods independent of the interactive educational system.

4.1.2 Review Correctness Prediction

Assume that a student incorrectly responds to an exercise e_{rev} and receives corresponding feedback. The goal of Review Correctness prediction (RC) is to predict whether a student will be able to respond to the exercise e_{rev} correctly if they encounter the exercise again. The significance of this AIEd task is that it can assess the educational effect of an exercise to a particular student in a specific situation. In particular, the correctness probability estimated by this task represents the student's expected marginal gain in knowledge as they go through some learning process. For example, if the correctness probability is high, it is likely that the student will obtain relevant knowledge in the future even if their initial response was incorrect.

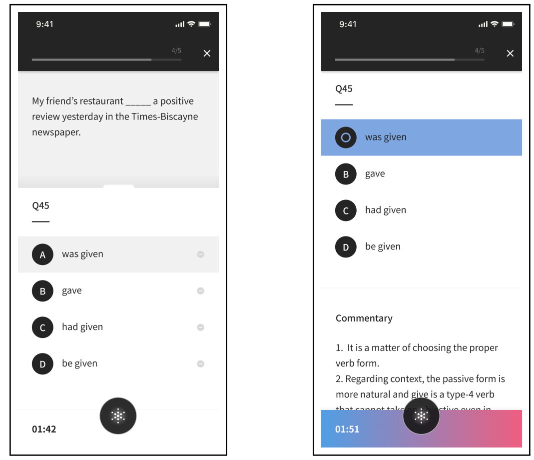

Figure 8: User Interface of *Santa*

Table 1: Time Limits

Part	1 ∼ 4	5	6	7
Time limit (sec)	audio duration + 8	25	50	55

4.2 Dataset

We use the public *EdNet* dataset obtained from *Santa*, a mobile AI tutoring service for TOEIC Listening and Reading Test preparation [4]. The test consists of two timed sections named Listening Comprehension (LC) and Reading Comprehension (RC) with a total of 100 exercises, and 4 and 3 parts respectively. The final test score ranges from 10 to 990 in steps of 5. Once a user solves each exercise, *Santa* provides educational feedback to their responses including explanations and commentaries on exercises. *EdNet* is the collection of user interactions of multiple-choice exercises collected over the last four years. The main features of the user-exercise interaction data consists of six columns: user id, exercise id, user response, exercise part, received time and time taken. We describe each column. Firstly, the user (resp. exercise) ID identifies each unique user (resp. exercise). The user response is recorded as 1 if the user response is correct and 0 otherwise. Exercise part is the part of the exam that the exercise belongs to. Finally, the absolute time when the user received the exercise and the time taken by the user to respond are recorded. In the dataset, 627,347 users solved more than one problem. The size of the exercise set is 16,175. The total row count of the dataset is 72,907,005.

4.2.1 Dataset for Pre-training

For pre-training, we first reconstruct the interaction timeline of each user by gathering the responses of a specific user in increasing chronological order. For each interaction I_t, the assessment value `response_correctness`$_t$ is the recorded user response correctness and `timeliness`$_t$ is recorded as 1 if the user responded under the time limits recommended by TOEIC experts (Table 1). We exclude the interactions

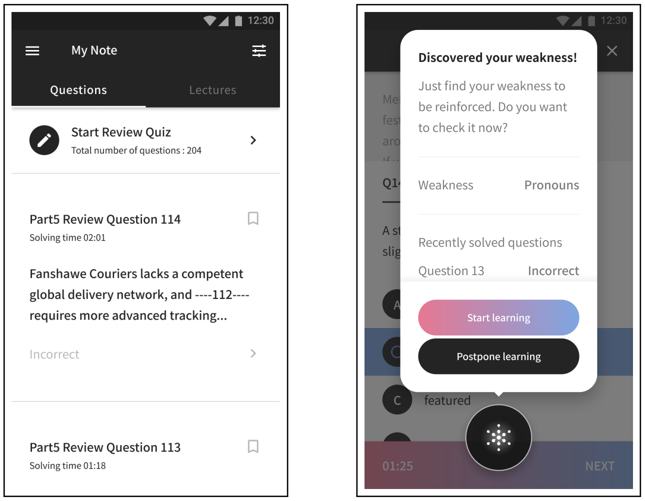

Figure 9: Review System of *Santa*

of users involved in any of the downstream tasks for pre-training to preemptively avoid data leakage. After processing, the data consists of 251,989 users with a total of 49,944,086 interactions.

4.2.2 Dataset for Label-Scarce Educational Problems

For exam score prediction, we aggregate the real TOEIC scores reported by users of *Santa*. The reports are scarce in number because a user has to register, take the exam and report the score at their own expense. To collect this data, *Santa* offered a small reward to users in exchange for reporting their score. A total of 2,594 score reports were obtained over a period of 6 months, which is considerably fewer than the number of exercise responses (72 million). For our experiment, we divide the data into a training set (1,302 users, 1815 labels), validation set (244 users, 260 labels), and test set (466 users, 519 labels).

For review correctness prediction, we look over each student's timeline and find exercises that have been solved at least twice. That is, if an exercise e appears more than once in a student interaction sequence $I_1 = (e_1, r_1), I_2 = (e_2, r_2), \ldots, I_T = (e_T, r_T)$, we find the first two interactions I_i and I_j $(i < j)$ with the same exercise e. The sequence of interactions $I_1, I_2, \cdots, I_i$ until they encounter e for the first time is taken as the input. The assessment $\texttt{response\_correctness}_j$ for the next encounter I_j is taken as the label. The total of 1,084,413 labeled sequence are generated after pre-processing. For our experiment, we divided the data into a training set (28,016 users, 747,222 labels), validation set (4,003 users, 112,715 labels), and test set (8,004 users, 224,476 labels).

4.3 Setup

4.3.1 Models

Our model consists of two encoder blocks ($N = 2$) with a latent space dimension of 256 ($d_{model} = 256$). The model takes 100 interactions as input. For comparison, the following pre-training methods are also applied to respective encoder networks with the same architecture. Since the existing pre-training methods embed the content of each exercise, we replace the embedding of `exercise_id` in our model with respective exercise embedding for fine-tuning.

- *Word2Vec* [21], a standard word embedding model, is used in [14] in generating exercise embeddings as follows. The model is pre-trained on the Google News dataset (∼100 billion words). The embedding assigned to each exercise is the average of embedding vectors of all words appearing in the exercise description. Embedded vectors are not fine-tuned.
- *BERT* [7] is a state-of-the-art pre-training method featuring the Transformer architecture trained on a masked language modeling objective. As above, we embed each exercise by averaging the representation vectors of words in the description without fine-tuning.
- Results for *QuesNet* [30] have been produced without using image and meta-data embeddings as the exercises used in our experiment consist only of text. We follow the architecture (Bi-directional LSTM followed by self-attention) and pre-training tasks (Holed Language Model and Domain-Oriented Objective) suggested in the original paper.

4.3.2 Metrics

We use the following evaluation metrics to evaluate model performance on each downstream task. For exam score prediction, we compute the Mean Absolute Error (MAE), the average of differences between the predicted exam scores and the true values. For review correctness prediction, we use accuracy (ACC), the proportion of correct predictions out of all predictions.

4.3.3 Training Details

The strategy we use for finding the optimal model parameters is the following. For pre-training, the model weights are first initialized with a Xavier uniform distribution and trained with $n_{pre} = 100$ epochs. After each i'th epoch, the model parameters are stored as $\mathcal{P}_i$. Then we fine-tune each $\mathcal{P}_i$ for a downstream task for a specified number $n_{down} = 100$ of epochs. Likewise, the model after j epochs of fine-tuning is stored as $\mathcal{D}_{i,j}$. Among all downstream task models $\mathcal{D}_{i,j}$ with $1 \leq i \leq n_{pre}$ and $1 \leq j \leq n_{down}$, the model with the best result on the validation set is chosen and evaluated with the test set. We use the Adam optimizer with hyperparameters $lr = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, epsilon = 1 - e8$. The batch size is 256 for pre-training and 128 for fine-tuning.

The labels available for exam score prediction are scarce. To alleviate this, we apply the following data augmentation when fine-tuning our model. Given the original interaction sequence with a score label, we select each element with 50% probability to generate a subsequence with the same score label. We train on these subsequences.

4.4 Experimental Results

4.4.1 Model Performance

Experimental results from the 5 different pre-trained models on the two aforementioned downstream tasks are shown in Table 2. In all downstream tasks, our model with Assessment Modeling outperforms the other models. Compared to

Table 2: Experimental results

	ES	RC
Without pre-train	64.37	0.648
Word2Vec	77.93	0.649
BERT	70.19	0.646
QuesNet	64.00	0.642
Assessment Modeling	**49.84**	**0.656**

Table 3: Assessment Modeling Tasks

Labels Pre-trained	ES	RC
`correctness`	54.28	**0.657**
`timeliness`	61.98	0.648
`correctness` + `timeliness`	**49.84**	0.656

the best model without pre-training, MAE for exam score prediction is reduced by 14.53 and ACC for review correctness is improved by 0.008 points. Our model also outperforms other pre-trained models, supporting our claim that Assessment Modeling is more suitable for Knowledge Tracing tasks than content-based pre-training approaches. This shows the importance of choosing pre-training tasks relevant to the downstream tasks.

4.4.2 *Effect of Assessment Modeling Labels*

Recall that our model is pre-trained to predict the whole assessment

$$A_t = (\texttt{response\_correctness}_t, \texttt{timeliness}_t).$$

We demonstrate the importance of pre-training on multiple assessments by comparing our model with two variations pre-trained to predict only one of $\texttt{response\_correctness}_t$ and $\texttt{timeliness}_t$. The results are shown in Table 3. For exam score prediction, the results show that it is the best to pre-train the model to predict the whole assessment A_t. This is because the exam score distribution is related not only to the student's answers but also the time they spent on each exercise. However, for review correctness prediction, the model only predicts the correctness of the user's response and does not care about timeliness. This explains why the model pre-trained to predict only the correctness of user's response performs slightly better than the model pre-trained to predict both components of assessments.

4.4.3 *Effect of Masking Positions*

As we mentioned in Section 3, we used a BERT-like masking method for pre-training to represent bi-directional information. We compared this with other possible masking approaches - masking the first 60% of exercises (Front) or the last 60% of exercises (Back) - and the results are shown in Table 4. For exam score prediction, our random masking approach is more effective than masking methods with fixed positions. In case of review correctness prediction, there is no big difference between the three models, but the model with last 60% of interactions masked performs slightly better than others.

4.4.4 *Ablation Study*

Table 4: Masking Positions

Masking Positions	ES	RC
Front	50.26	0.656
Back	51.91	**0.657**
Random	**49.84**	0.656

Table 5: Ablation Study. Rows in (A) show the proportion C of masked interactions. Rows in (B) show the masking rate M. Rows in (C) show the parameters for model architecture (number N of encoder blocks and dimension d_{model} of latent space)

	C	M	N	d_{model}	ES	RC
Base	0.6	1.0	2	256	**49.84**	0.656
(A)	0.2				53.87	0.655
	0.4				51.79	0.655
	0.8				52.01	**0.657**
(B)		0.0			52.25	0.655
		0.5			52.51	0.655
(C)			1	128	50.79	**0.657**
			4	512	51.85	0.653

We conduct several ablation experiments to understand how each property of our model affects the model's performance. First, we observe the effect of the hyper-parameters for masking: the proportion of masked interactions C and the masking rate M. Then, we varied the size of the model, the number of layers N, and the dimension of model d_{model}. As a result, the model with $C = 0.6$, $M = 1.0$, $N = 2$, and $d_{\text{model}} = 256$ performs the best.

5. CONCLUSION

In this paper, we introduced Assessment Modeling, a class of fundamental pre-training tasks for Interacvive Educational Systems. Our experiments show the effectiveness of Assessment Modeling as pre-training tasks for label-scarce educational problems including exam score and review correctness prediction. Future works may include assessments beyond correctness and timeliness of the responses and investigate further label-scarce educational problems. Investigation on these subjects is ongoing.

6. REFERENCES

[1] A. Almasri, E. Celebi, and R. S. Alkhawaldeh. Emt: Ensemble meta-based tree model for predicting student performance. *Scientific Programming*, 2019, 2019.

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Y. Choi, C. Y.-I. Chiu, and D. Sontag. Learning low-dimensional representations of medical concepts. *AMIA Summits on Translational Science Proceedings*, 2016:41, 2016.

[4] Y. Choi, Y. Lee, D. Shin, J. Cho, S. Park, S. Lee, J. Baek, C. Bae, B. Kim, and J. Heo. Ednet: A large-scale hierarchical dataset in education, 2019.

[5] A. T. Corbett and J. R. Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*,

4(4):253–278, 1994.

[6] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu. Boosting for transfer learning. In *Proceedings of the 24th international conference on Machine learning*, pages 193–200. ACM, 2007.

[7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[9] M. Ding, Y. Wang, E. Hemberg, and U.-M. O'Reilly. Transfer learning using representation learning in massive open online courses. In *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, pages 145–154. ACM, 2019.

[10] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio. Why does unsupervised pre-training help deep learning? *Journal of Machine Learning Research*, 11(Feb):625–660, 2010.

[11] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1243–1252. JMLR. org, 2017.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] Z. Huang, Q. Liu, E. Chen, H. Zhao, M. Gao, S. Wei, Y. Su, and G. Hu. Question difficulty prediction for reading problems in standard tests. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[14] Z. Huang, Y. Yin, E. Chen, H. Xiong, Y. Su, G. Hu, et al. Ekt: Exercise-aware knowledge tracing for student performance prediction. *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[15] X. J. Hunt, I. K. Kabul, and J. Silva. Transfer learning for education data. In *KDD Workshop*, 2017.

[16] Z. Iqbal, J. Qadir, A. N. Mian, and F. Kamiran. Machine learning based student grade prediction: A case study. *arXiv preprint arXiv:1708.08744*, 2017.

[17] Y. Lee, Y. Choi, J. Cho, A. R. Fabbri, H. Loh, C. Hwang, Y. Lee, S.-W. Kim, and D. Radev. Creating a neural pedagogical agent by jointly learning to review and assess, 2019.

[18] Q. Liu, Z. Huang, Z. Huang, C. Liu, E. Chen, Y. Su, and G. Hu. Finding similar exercises in online education systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1821–1830. ACM, 2018.

[19] Q. Liu, S. Tong, C. Liu, H. Zhao, E. Chen, H. Ma, and S. Wang. Exploiting cognitive structure for adaptive learning. *arXiv preprint arXiv:1905.12470*, 2019.

[20] R. Luckin, W. Holmes, M. Griffiths, and L. B. Forcier. Intelligence unleashed: An argument for ai in education. 2016.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[22] S. Morsy and G. Karypis. Cumulative knowledge-based regression models for next-term grade prediction. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 552–560. SIAM, 2017.

[23] S. Pandey and G. Karypis. A self-attentive model for knowledge tracing. *arXiv preprint arXiv:1907.06837*, 2019.

[24] A. P. Patil, K. Ganesan, and A. Kanavalli. Effective deep learning model to predict student grade point averages. In *2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–6. IEEE, 2017.

[25] C. Piech, J. Bassen, J. Huang, S. Ganguli, M. Sahami, L. J. Guibas, and J. Sohl-Dickstein. Deep knowledge tracing. In *Advances in neural information processing systems*, pages 505–513, 2015.

[26] S. Schneider, A. Baevski, R. Collobert, and M. Auli. wav2vec: Unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862*, 2019.

[27] L. Studer, M. Alberti, V. Pondenkandath, P. Goktepe, T. Kolonko, A. Fischer, M. Liwicki, and R. Ingold. A comprehensive study of imagenet pre-training for historical document image analysis. *arXiv preprint arXiv:1905.09113*, 2019.

[28] C. Sung, T. I. Dhamecha, and N. Mukhi. Improving short answer grading using transformer-based pre-training. In *International Conference on Artificial Intelligence in Education*, pages 469–481. Springer, 2019.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[30] Y. Yin, Q. Liu, Z. Huang, E. Chen, W. Tong, S. Wang, and Y. Su. Quesnet: A unified representation for heterogeneous test questions. *arXiv preprint arXiv:1905.10949*, 2019.

[31] J. Zhang, X. Shi, I. King, and D.-Y. Yeung. Dynamic key-value memory networks for knowledge tracing. In *Proceedings of the 26th international conference on World Wide Web*, pages 765–774. International World Wide Web Conferences Steering Committee, 2017.